Deep Learning (CSE641/ECE555)
Mid-Sem (20 Marks) (Duration 60 min)

Name .....
Roll No. .....

Beginner: 10 Marks

1. The loss function of a VAE includes a term for: B

   (a) Overfitting the decoder network. (b) Matching the reconstructed data to the original. (c) Ignoring irrelevant information in the latent space. (d) All of the above.

2. VAEs offer advantages over standard autoencoders because they: B

   (a) Can only reconstruct binary data. (b) Allow for more complex and diverse reconstructions. (c) Require less training data. (d) Are always guaranteed to produce perfect reconstructions.

3. Which of the following statements about pooling layers in CNNs is INCORRECT? D

   (a) They can help reduce the dimensionality of the data. (b) They can introduce some degree of invariance to translations. (c) They can be used to capture spatial relationships between features. (d) They do not learn any parameters themselves.

4. In a residual network (ResNet), what is the main purpose of the skip connections? A

   (a) They allow for deeper networks to be trained more easily. (b) They introduce randomness into the network for better generalization. (c) They compress the activations in the network for memory efficiency. (d) They act as a form of attention mechanism, focusing on important features.

5. When choosing an appropriate activation function for the convolutional layers in a CNN, which of the following factors should be given the HIGHEST priority? C

   (a) The computational efficiency of the function. (b) The ability of the function to handle negative inputs. (c) The function's non-linearity to learn complex relationships. (d) Whether the function has a derivative for backpropagation.

6. Which of the following statements about the perceptron convergence theorem (PCT) is true? B

   (a) PCT guarantees convergence to the global minimum of the loss function. (b) PCT guarantees convergence to a solution if the data is linearly separable. (c) PCT guarantees convergence regardless of the initial weights. (d) PCT guarantees convergence for any activation function.

7. During training a CNN for image classification, what does the following term in the loss function represent: $\lambda \times ||W||^2, \lambda \in (0, 1)$ C

   (a) It encourages the network to learn more complex filters. (b) It penalizes the model for activation values exceeding a certain threshold. (c) It helps prevent overfitting by reducing the model's complexity. (d) It improves the interpretability of the learned filters.

8. Shyam devised an activation function ($f(x) = x * \tanh(\log(1 + \exp(x)))$). When compared to the ReLU activation function, what is a potential advantage of this activation function? C

   (a) It enforces sparsity in the activations, reducing the number of weights to update. (b) It has a unbounded output range like sigmoid, simplifying normalization techniques. (c) It offers a smoother derivative around x = 0, potentially improving convergence. (d) It allows for dynamic weight scaling during training, similar to weight decay.

9. Consider an Inception module with two branches: `B1:1x1 convolution with 64 filters`, `B2:5x5 convolution with 32 filters`. The input to the Inception module has a size of 28 x 28 x 192. After applying the convolutions in each branch, what is the output size of each branch (assuming stride 1 and no padding)? A

   (a) B1: 28 x 28 x 64, B2: 24 x 24 x 32 (b) B1: 28 x 28 x 64, B2: 28 x 28 x 32 (c) B1: 26 x 26 x 64, B2: 24 x 24 x 32 (d) B1: 27 x 27 x 64, B2: 23 x 23 x 32

10. The β-VAE is a variational autoencoder that incorporates a hyperparameter β. How does β affect the model's behavior? A

(a) Higher β encourages latent representation with higher entropy. (b) Lower β leads to poor reconstruction accuracy. (c) β controls convergence of VAE. (d) β controls latent representations dimensions.

Intermediate: 6 Marks

1. Consider a neural network with an input layer, one hidden layer, and an output layer. The input layer has 3 neurons, the hidden layer has 5 neurons, and the output layer has 1 neuron. If all the activation functions are sigmoid functions, what is the maximum number of linear hyperplanes the network can create? 1 Mark C

   (a) 3 (b) 5 (c) 20 (d) 21

2. Can we use transposed convolution with fractional strides. If so, how does using a fractional stride (e.g., 0.5) in a transposed convolution layer with appropriate padding affect the output size compared to using an integer stride? 1 Mark B

   (a) No, Code implimentation not possible (b) Yes, Increases the output size compared to using the same integer stride. (c) No, Does not affect output size compared to using the same integer stride. (d) Yes, Introduces checkerboard artifacts in the output due to the non-integer stride placement.

3. Gopi wants to use Xavier initialization on a network with n_{in} input and n_{out} output neurons with ReLU activation function. Write the initialization formula for the weights and explain it. 1 Mark

   DL Lecture 7 slide 32

   $$\mathcal{U}(-sqrt(6/(n_in + n_out)), sqrt(6/(n_in + n_out)))$$

4. What makes the approximation capabilities of neural networks different than something like, say, Fourier series? 1 Mark

   Non-linear DNN can approximate any function due to nonlinearity, but Fourier can't when certain conditions are not fulfilled, such as discontinuities. DNN has a better approximation rate, given a carefully designed architecture.

5. Why does the activation function for a single hidden layer MLP have to be non-polynomial? 2 Mark

   If the activation function is polynomial, the entire network (input to output mapping) collapses into a polynomial function of the inputs. The Universal Approximation Theorem (Cybenko, 1989; Hornik, 1991) states that a single hidden-layer MLP with a nonlinear activation function can approximate any continuous function on a compact domain arbitrarily well. However, this theorem does not hold if the activation function is a polynomial. The reason is that polynomials are not dense in the space of continuous functions—meaning that some continuous functions cannot be approximated by polynomials. A well-known counterexample is the indicator function of a set, which cannot be expressed using polynomials due to Runge's phenomenon (oscillatory behavior of polynomial approximations).

Advanced: 4 Marks

1. Grad-CAM++ is an improvement over Grad-CAM visualization technique that provides better localization by considering higher-order gradient terms. The Grad-CAM++ activation map corresponding to the importance weight α_k^c for feature map A_k is computed as:

$$\alpha_k^c = \sum_{i,j} w_k^{ij} \frac{\partial^2 f^c}{\partial (A_k^{ij})^2} \qquad w_k^{ij} = \frac{\frac{\partial f^c}{\partial A_k^{ij}}}{\sum_{p,q} \frac{\partial f^c}{\partial A_k^{pq}}} \qquad L^c_{\text{Grad-CAM++}} = \text{ReLU}\left(\sum_k \alpha_k^c A_k\right)$$

   where w_k^{ij} is the pixel-wise weighting factor.

   Answer the following with explanation:

(a) Why is ReLU Necessary?

(b) What happens when the class scores f^c does not change significantly with respect to variations in the feature maps?

(c) Will GradCAM++ be better than GradCAM if the model's gradients are noisy or unstable?

(d) Discuss the impact of depth and width on GradCAM++ visualization?

The negative values in Grad-CAM correspond to regions where decreasing the activation would increase the class score. These regions do not contribute positively to the class and should be ignored in visualization. Applying ReLU ensures that only positive activations are highlighted in the heatmap.
If the class score does not change significantly with respect to variations in the feature maps, second-order gradients will be close to zero, making Grad-CAM++ ineffective.
Grad-CAM++ relies on second-order gradients, which are more sensitive to noise. If the model has unstable gradients (e.g., due to poor training or adversarial examples), Grad-CAM++ may highlight irrelevant regions. Example: In adversarially perturbed images, where small pixel changes drastically alter predictions, Grad-CAM++ might amplify the noise rather than provide meaningful localization.
The second-order gradients become unstable due to vanishing gradient problems in deep layers. Layer aggregation makes it hard for Grad-CAM++ to correctly attribute relevance, leading to diffused or incorrect visualizations. In very wide architectures Grad-CAM++ suffers from gradient saturation across a large number of feature maps, which might cause the attention map to become too spread out, leading to less focused visualizations. Grad-CAM++ needs diverse activation maps to properly distribute importance. In shallow networks, feature maps are too simple, and higher-order gradients become nearly redundant.
Shallow or extremely deep networks? $\rightarrow$ Use Grad-CAM Moderate-depth models with diverse feature maps? $\rightarrow$ Use Grad-CAM++